NONLINEAR OPTIMIZATION

Qingsha Cheng 程庆沙

Notation and Repetition of Necessary Mathematical Concepts

Notation used throughout the course

Vectors and matrices

Linear systems

Derivatives; gradient; Hessian; Jacobian

Taylor expansion

Approximation of derivatives using finite differences

Notation

Functions:

$$f : X \to Y$$

Function name
(typically *f*, *g*, *h*)

Set of values
(counter-domain)

Function domain

Symbol R will be used to denote the set of real numbers

R^n denotes the n Cartesian product of n copies of R

R_+ denotes the set of all positive real numbers

Notation

Vectors:

Transposition

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \end{bmatrix}^T$$

Vector name
(typically ***x***, ***y***, ***z***)

Vector components
(elements)

Notation

Matrices:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = \left[a_{ij} \right]_{\substack{i=1,\dots,m \\ j=1,\dots,n}}$$

Matrix name
(typically *A*, *B*, *C*)

Matrix entries

Identity matrices (i.e., square matrices having ones on the diagonal and zeros elsewhere) are denoted as I or I_n (identity matrix of size n)

Remark: vectors are also matrices, however they are distinguished because of their importance in mathematics.

Vector and Matrix Operations

Matrix transposition:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \quad => \quad \boldsymbol{A}^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix}$$

We have: $\left(\boldsymbol{A}^T\right)^T = \boldsymbol{A}$

$\boldsymbol{A}$ is a symmetric matrix if $\boldsymbol{A}^T = \boldsymbol{A}$ or $a_{ij}=a_{ji}$

Vector and Matrix Operations

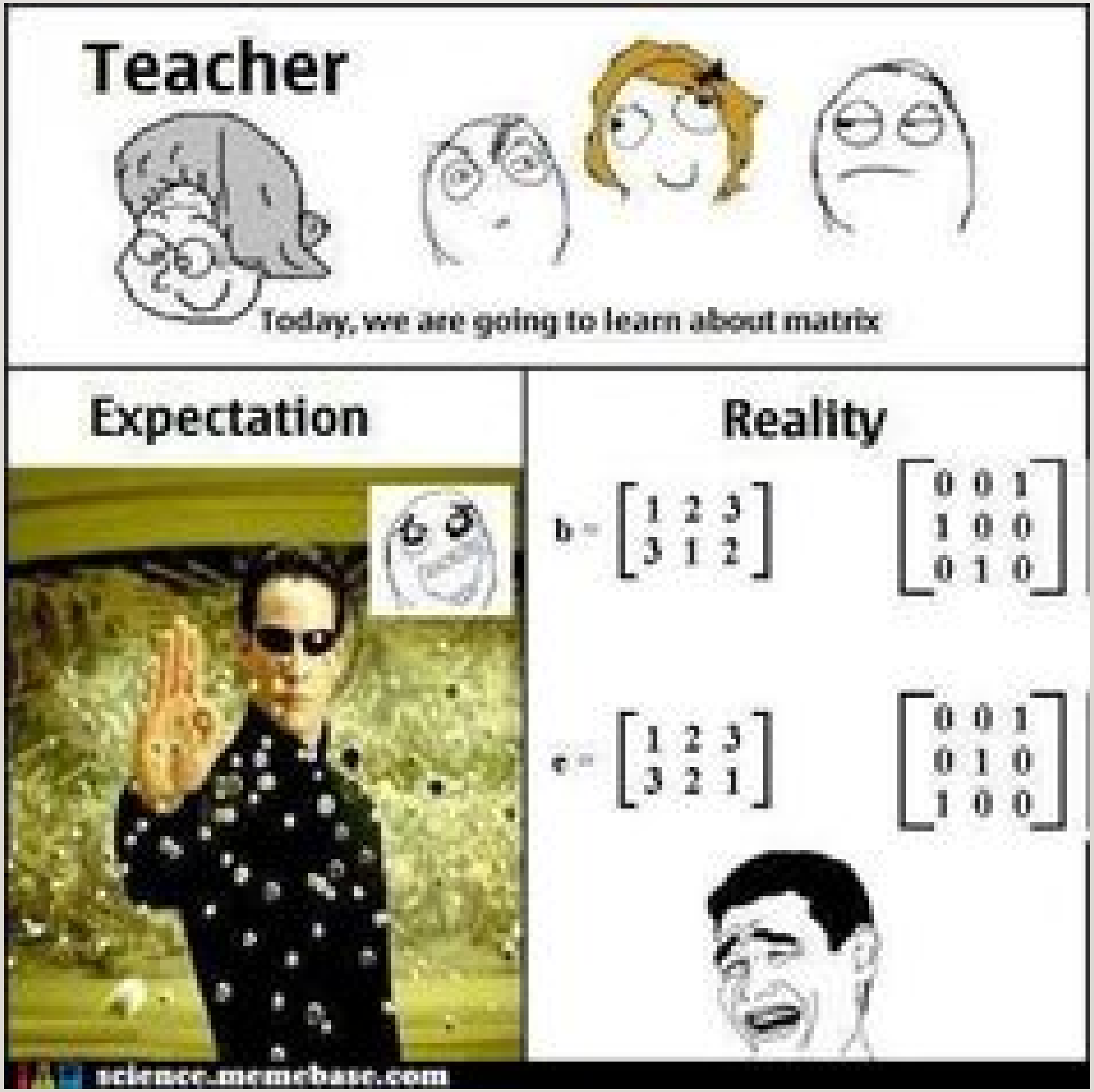
Matrix multiplication:

$$A_{m\times n}B_{n\times k} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & \cdots & b_{1k} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nk} \end{bmatrix} = \begin{bmatrix} c_{11} & \cdots & c_{1k} \\ \vdots & \ddots & \vdots \\ c_{m1} & \cdots & c_{mk} \end{bmatrix} = C_{m\times k}$$

where $c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$

In general $AB \neq BA$!!! (Commutative law doesn't apply for matrix multiplication)

We have $(AB)^T = B^T A^T$ (in particular $AB = BA$ for symmetric matrices)


Teacher
Today, we are going to learn about matrix
Expectation
Reality
science.memebase.com

Determinant

Determinant is a function defined for $n \times n$ matrices that can be calculated along ith row (or column) as follows (Laplace's formula)

$$\det(A) = \sum_{j=1}^{n} a_{ij} (-1)^{i+j} M_{ij}$$

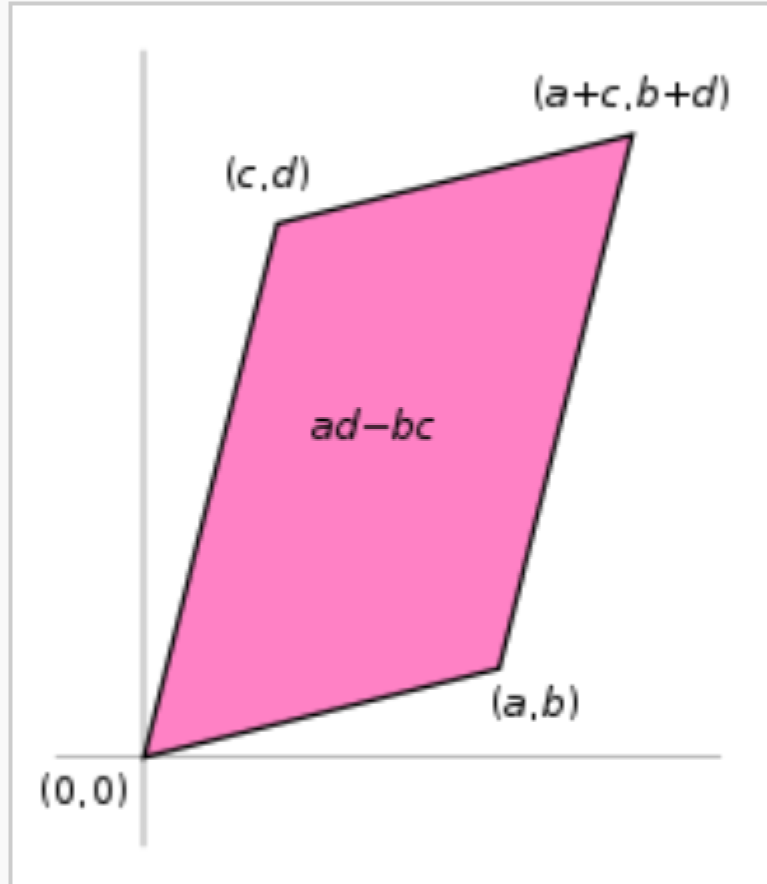
where M_{ij} is the determinant of the matrix that results from A by removing ith row and jth column (determinant of 1×1 matrix is equal to its entry).

Remarks:

Laplace's formula is a very inefficient way of calculating determinant, which cannot be used for large matrices (in practice for $n > 5$ to 8) because of exponential complexity (number of multiplications required is proportional to $n!$)

Determinant

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc.$$



The area of the parallelogram is the absolute value of the determinant of the matrix formed by the vectors representing the parallelogram's sides.

Determinant

There are various formulas for calculating determinants of low-dimensional matrices:

1. $\det(A) = a_{11}$ for 1×1 matrix

2. $\det(A) = a_{11}a_{22} - a_{12}a_{21}$ for 2×2 matrix

3. $$\det(A) = a_{11}a_{22}a_{33} + a_{13}a_{21}a_{32} + a_{12}a_{23}a_{31} \\ - a_{13}a_{22}a_{31} - a_{11}a_{23}a_{32} + a_{12}a_{21}a_{33}$$

In practice, determinants are calculated through various matrix decompositions, which is much more efficient than using a Laplace's formula

Invertible Matrices

$n \times n$ matrix A is called invertible if there exists an $n \times n$ matrix B such that $AB = BA = I_n$

A square matrix that is not invertible is called ***singular***

A square matrix A is ***singular*** if and only if $\det(A) = 0$

(Inefficient) analytic calculation of A^{-1}:

$$A^{-1} = \frac{1}{\det(A)}\left((-1)^{i+j} M_{ij}\right)^T$$

where M_{ij} is the determinant of the matrix that results from A by removing ith row and jth column

Various Properties

Various properties:

$\det(AB) = \det(A)\det(B)$

$\det(A^{-1}) = \det(A)^{-1}$

$\det(A^T) = \det(A)$

$\det(kA) = k^n\det(A)$

$(A^T)^T = A$

$(A+B)^T = A^T + B^T$

$(kA)^T = kA^T$

$(A^{-1})^{-1} = A$

$(AB)^T = B^TA^T$

$(A^T)^{-1} = (A^{-1})^T$

$(AB)^{-1} = B^{-1}A^{-1}$

Inner Product and Norm

Inner product and vector/matrix norm*:

Multiplication of vectors of the form x^Ty is a so-called *inner product* of x and y; we have $x^T y = y^T x$

Norm of a vector x, $||x||$ is defined as $\| x \| = \left\| [x_1 \;\; x_2 \;\; \ldots \;\; x_n]^T \right\| = \sqrt{\sum_{j=1}^{n} x_j^2}$

We have $x^T x = ||x||^2$

Norm of a matrix A, $||A||$ is defined as $\| A \| = \left\| [a_{ij}]_{i,j=}^{n} \right\| = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^2}$

* The norm definitions given here are so-called *L*-square norms and these are not the only possible norms of vectors/matrices used in practice.

Positive-Definite Matrices

A square $n \times n$ matrix $\boldsymbol{A}$ is called positive-definite if for all non-zero vectors $\boldsymbol{x}$ Î R^n we have $x^T\boldsymbol{A}x > 0$ (definitions of positive-semidefinite and negative-definite matrices can be obtained by replacing $>$ by $\geq$ and $<$, respectively)

Eigenvectors and **eigenvalues**: eigenvectors x of a square matrix A are defined as those vectors which, when multiplied by $\boldsymbol{A}$, result in a simple scaling λ of x, i.e., $\boldsymbol{A}x = \lambda x$. λ is called the eigenvalue corresponding to eigenvector x.

Fact: a symmetric matrix $\boldsymbol{A}$ is positive-definite if and only if its all eigenvalues are positive

Eigen Value and Eigen Vector

In Matlab

[V,D] = eig(X)

produces a diagonal matrix D of eigenvalues and a
full matrix V whose columns are the corresponding eigenvectors so that X*V = V*D.

LU Decomposition of a Square Matrix

A square matrix $\boldsymbol{A}$ can be decomposed into a lower-triangular matrix L and upper triangular matrix U so that $A=LU$, i.e.

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ l_{n1} & \cdots & l_{nn} \end{bmatrix} \cdot \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & u_{nn} \end{bmatrix} = LU$$

LU decomposition is a very convenient tool because operations on triangular matrices (e.g., inversion, finding determinant) are easy

In Matlab:

[L,U] = lu(A) stores an upper triangular matrix in U and a "psychologically lower triangular matrix" (i.e. a product of lower triangular and permutation matrices) in L, so that A = L*U. A can be rectangular.

LU Decomposition of a Square Matrix

Applications:

1.Solving linear equations: $Ax = LUx = b$: first solve $Ly = b$ for y, then $Ux = y$ for x

2.Inverse matrix: $A^{-1}=U^{-1}L^{-1}$

3.Determinant: $\det(A) = \det(L)\det(U) = u_{11}\cdot u_{22}\cdot \ldots u_{nn}$

In Matlab [l u]=lu(A);

Linear Independence and Rank of a Matrix

A set of vectors v_1, v_2, ..., v_k, is linearly independent if none of them can be written as a linear combination of other vectors in the set, i.e., the only solution to $\alpha_1 v_1+\alpha_2 v_2+\ldots+\alpha_k v_k = 0$ is $\alpha_1 = \alpha_2 = \ldots = \alpha_k = 0$

Rank of a matrix A is the maximal number of linearly independent columns (rows) of A

We have: $\mathrm{rank}(A_{m\times n}) \leq \min(m,n)$

Fact: vectors v_1, v_2, ..., v_k, are linearly independent if and only if the matrix A whose columns are vectors v_1 to v_k has rank k

Square $n\times n$ matrix A is invertible if and only if A has rank n, which is if an only if columns (rows) of A are linearly independent

A little bit Matlab Fun

Systems of Linear Equations

A general system of *m* linear equations with *n* unknowns can be written as:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m
\end{aligned}
$$

or, in a matrix form as $Ax = b$

where

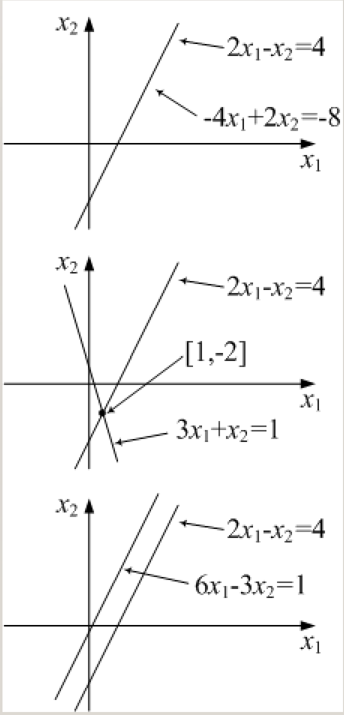
$$
A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \qquad b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}
$$

Systems of Linear Equations

A linear system may behave in any one of the three ways:

1. The system has infinitely many (parameterized) solutions
2. The system has a single unique solution
3. The system has no solutions

Geometrical interpretation



Case 1: $$\begin{aligned} 2x_1 - x_2 &= 4 \\ -4x_1 + 2x_2 &= -8 \end{aligned}$$ solutions: any x_1, $x_2 = 2x_1-4$

Case 2: $$\begin{aligned} 2x_1 - x_2 &= 4 \\ 3x_1 + x_2 &= 1 \end{aligned}$$ solution: $x_1 = 1$, $x_2 = -2$

Case 3: $$\begin{aligned} 2x_1 - x_2 &= 4 \\ 6x_1 - 3x_2 &= 1 \end{aligned}$$ no solutions

Systems of Linear Equations

Case $m = n$ (n equations with n unknown):

We will assume that all linearly dependent equations are eliminated

Assume also that $b \neq 0$ (i.e., the right-hand side of the system is non-zero).

Then, system $Ax = b$ has a unique solution if and only if matrix A is non-singular (i.e., $\det(A) \neq 0$); the solution is given as $x = A^{-1}b$

Otherwise, system $Ax = b$ has no solutions

Systems of Linear Equations

A system of linear equations is *homogeneous* if the right-hand side of the system is equal 0, i.e., $b = 0$

Homogeneous system always has at least zero solution (also called trivial solution), obtained by assigning zero to each of the variables

In case of $m = n$, trivial solution is the only solution of the system if and only if matrix A is non-singular; otherwise, the system has infinite number of solutions

Systems of Linear Equations

Gauss elimination procedure for the case $m = n$:

Gauss elimination allows solving linear systems based on the observation that the system is invariant with respect to swapping equations, multiplication of equations by scalars as well as adding equations

Initial system:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= b_n \end{aligned}$$

Systems of Linear Equations

Elimination procedure (assume that all equations are linearly independent):

for each i from 1 to n-1 perform the following steps
    swap equations i to n so that the one with the highest value of $|a_{ji}|$ is set as equation i
    if $|a_{ji}| = 0$
        system has no solutions => terminate
    end
    for each j from i+1 to n perform the following steps
        add equation i multiplied $-a_{ji}/a_{ii}$ to equation j
    end
end
find solution using back-substitution

Note that the procedure is terminated if there is no non-zero a_{ji} at step i, which means that the matrix A is singular

Systems of Linear Equations

First elimination step (eliminate x_1 for equ. 2~n):

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1$$

$$(a_{22} - a_{12}a_{21} / a_{11})x_2 + \ldots + (a_{2n} - a_{1n}a_{21} / a_{11})x_n = b_2 - b_1 a_{21} / a_{11}$$

$$\vdots$$

$$(a_{n2} - a_{12}a_{n1} / a_{11})x_2 + \ldots + (a_{nn} - a_{1n}a_{n1} / a_{11})x_n = b_n - b_1 a_{n1} / a_{11}$$

Second elimination step (eliminate x_2 for equ. 3~n coefficients in first two equations have updated for simplicity):

$$a_{11}^{'}x_1 + a_{12}^{'}x_2 + \ldots + a_{1n}^{'}x_n = b_1^{'}$$

$$a_{22}^{'}x_2 + \ldots + a_{1n}^{'}x_n = b_2^{'}$$

$$(a_{33} - a_{23}^{'}a_{32} / a_{22}^{'})x_3 + \ldots + (a_{3n} - a_{2n}^{'}a_{32} / a_{22}^{'})x_n = b_3 - b_2^{'}a_{32} / a_{22}^{'}$$

$$\vdots$$

$$(a_{n2} - a_{23}^{'}a_{n2} / a_{22}^{'})x_3 + \ldots + (a_{nn} - a_{2n}^{'}a_{n2} / a_{22}^{'})x_n = b_n - b_2^{'}a_{n2} / a_{22}^{'}$$

Systems of Linear Equations

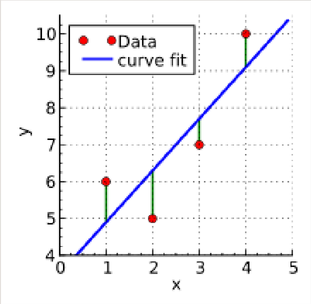
After *n*-1 steps we obtain a triangular system of the form:

$$
\begin{aligned}
\bar{a}_{11}x_1 + \bar{a}_{12}x_2 + \ldots + \bar{a}_{1n}x_n &= \bar{b}_1 \\
\bar{a}_{22}x_2 + \ldots + \bar{a}_{2n}x_n &= \bar{b}_2 \\
&\vdots \\
\bar{a}_{nn}x_n &= \bar{b}_n
\end{aligned}
$$

Solution is obtained using back-substitution (starting from x_n and finishing on x_1):

$$x_i = \left(\bar{b}_i - \sum_{j=i+1}^{n} \bar{a}_{ij} x_j\right) \bar{a}_{ii}^{-1}$$

Systems of Linear Equations



Case $m > n$ (over-determined systems):

We again assume that all linearly-dependent equations are eliminated and that the system is not homogeneous

Then, the system has no solutions

Important case: rank(A) = n; it is then possible to find an approximate solution, which is the best solution in a least squares sense, i.e., it realizes $\min_x \|Ax - b\|$, and which is given by a formula:

$$x = \left(A^T A\right)^{-1} A^T b$$

If rank(A) < n, then there is infinitely many least squares solutions

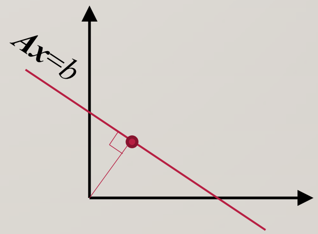
Systems of Linear Equations

Case $m < n$ (under-determined systems):

We again assume that all linearly-dependent equations are eliminated

Important case: rank(A) = m; the system $Ax = b$ has then infinite number of solutions, however, it is possible to find the one which is the minimum in a least squares sense; in particular

$$x = A^T \left(AA^T \right)^{-1} b$$



realizes $x = \min\{||y|| : Ay = b\}$

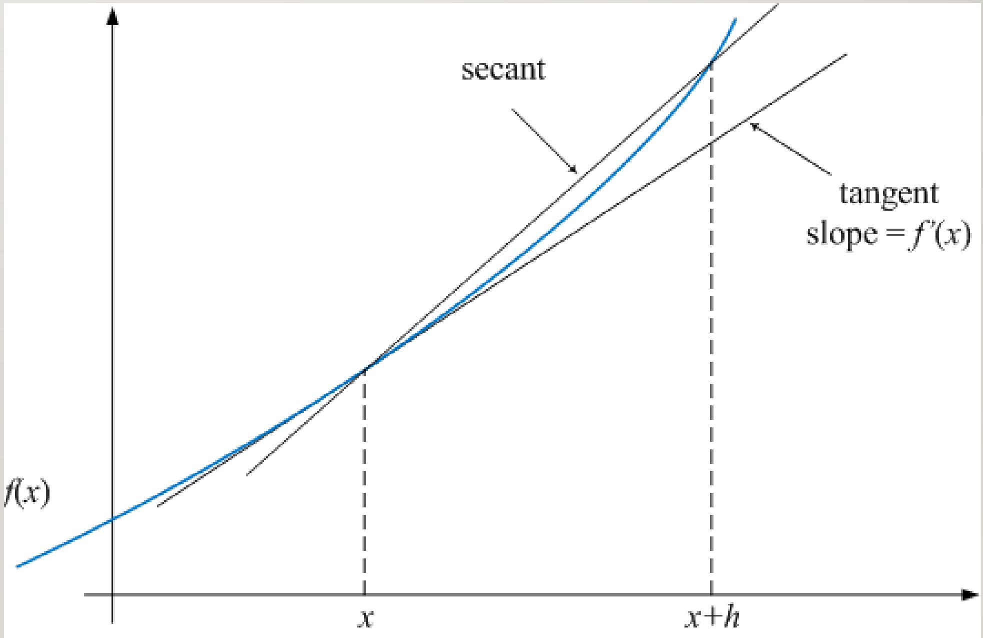
If rank(A) < m, then the system has no solutions, however, there is infinite number of approximate solutions which are best in a least squares sense (see: over-determined systems)

Derivatives: Scalar Function of a Single Variable

Derivative of a scalar function of a single variable $f : R \to R$:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Geometric interpretation:



Notation: f' (f'', f''', for second-, third-order derivative, etc.)

Alternative notation: $\frac{df}{dx}$ $\left(\frac{d^2 f}{dx^2}, \frac{d^3 f}{dx^3}\right)$ or $f^{(1)}$ ($f^{(2)}$, $f^{(3)}$)

Derivatives: Scalar Function of a Single Variable

Example: Calculate derivative of a monomial x^3 by definition:

$$(x^3)' = \lim_{h\to 0}\frac{(x+h)^3 - x^3}{h} = \lim_{h\to 0}\frac{x^3 + 3x^2h + 3xh^2 + h^3 - x^3}{h}$$

$$= \lim_{h\to 0}\frac{3x^2h + 3xh^2 + h^3}{h} = \lim_{h\to 0}\left(3x^2 + 3xh + h^2\right) = 3x^2$$

Formulas for calculating derivatives of common elementary functions:

$$(x^r)' = rx^{r-1} \qquad (e^x)' = e^x \qquad (\ln(x))' = 1/x$$

$$(\sin(x))' = \cos(x) \qquad (\cos(x))' = -\sin(x)$$

Rules of calculating derivatives:

$$(af + bg)' = af' + bg' \qquad (f/g)' = (f'g - fg')/g^2$$

$$(fg)' = f'g + fg' \qquad (f(g(x))' = f'(g(x))g'(x)$$

Derivatives: Scalar Function of Many Variables

Derivative of a scalar function of a vector variable $f : R^n \rightarrow R$:

$$\frac{\partial f}{\partial x_i}(\mathbf{x}) = \frac{\partial f}{\partial x_i}(x_1, x_2, ..., x_n) = \lim_{h \to 0} \frac{f(x_1, ..., x_i + h, ..., x_n) - f(x_1, ..., x_n)}{h}$$

Gradient: vector of partial derivatives of f with respect to all variables

$$f'(\mathbf{x}) = \nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial x_1}(\mathbf{x}) \ \ ... \ \ \frac{\partial f}{\partial x_n}(\mathbf{x}) \right]^T$$

Directional derivative in the direction of vector $\boldsymbol{v} = [v_1, ..., v_n]^T$:

$$D_v f(\mathbf{x}) = \lim_{h \to 0} \frac{f(\mathbf{x} + h\mathbf{v}) - f(\mathbf{x})}{h}$$

Notation of higher order partial derivatives: $\frac{\partial^2 f}{\partial x_i^2}, \frac{\partial^3 f}{\partial x_i^3}, ...$

Derivatives: Scalar Function of Many Variables

Mixed partial derivatives are denoted as $\dfrac{\partial^2 f}{\partial x_i \partial x_j}$

Second derivative of a scalar function of a vector variable $f : R^n \to R$ is called Hessian:

$$H_f(\mathbf{x}) = \nabla^2 f(\mathbf{x}) = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2}(\mathbf{x}) & \dfrac{\partial^2 f}{\partial x_1 \partial x_2}(\mathbf{x}) & \cdots & \dfrac{\partial^2 f}{\partial x_1 \partial x_n}(\mathbf{x}) \\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1}(\mathbf{x}) & \dfrac{\partial^2 f}{\partial x_2^2}(\mathbf{x}) & \cdots & \dfrac{\partial^2 f}{\partial x_2 \partial x_n}(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f}{\partial x_n \partial x_1}(\mathbf{x}) & \dfrac{\partial^2 f}{\partial x_n \partial x_2}(\mathbf{x}) & \cdots & \dfrac{\partial^2 f}{\partial x_n^2}(\mathbf{x}) \end{bmatrix}^T$$

Derivatives: Vector Function of Many Variables

Jacobian: if $\boldsymbol{f} = [f_1, f_2, \ldots, f_n]^T$, then the derivative can be expressed as a matrix called the Jacobian matrix of $\boldsymbol{f}$ at $\boldsymbol{x}$:

$$f\,'(\boldsymbol{x}) = J_f(\boldsymbol{x}) = \left[\frac{\partial f_i}{\partial x_j}(\boldsymbol{x})\right]_{ij} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\boldsymbol{x}) & \frac{\partial f_1}{\partial x_2}(\boldsymbol{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\boldsymbol{x}) \\ \frac{\partial f_2}{\partial x_1}(\boldsymbol{x}) & \frac{\partial f_2}{\partial x_2}(\boldsymbol{x}) & \cdots & \frac{\partial f_2}{\partial x_n}(\boldsymbol{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1}(\boldsymbol{x}) & \frac{\partial f_m}{\partial x_2}(\boldsymbol{x}) & \cdots & \frac{\partial f_m}{\partial x_n}(\boldsymbol{x}) \end{bmatrix}^T$$

Taylor Expansion: Scalar Function of a Single Variable

Function $f(x)$ that is infinitely differentiable in a neighborhood of x_0 can be expanded using Taylor power series as follows:

$$f(x_0) + \frac{f^{(1)}(x_0)}{1!}(x - x_0) + \frac{f^{(2)}(x_0)}{2!}(x - x_0)^2 + \frac{f^{(3)}(x_0)}{3!}(x - x_0)^3 + \dots$$

or, in a compact form

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n$$

Taylor Expansion: Scalar Function of a Single Variable

Error of a truncated series expansion:

$$f(x) = \sum_{n=0}^{k} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n + R_n(x)$$

where $R_n(x) = \dfrac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)^{n+1}$ for some ξ between x and x_0

We also have $|R_n(x)| \leq M_n \dfrac{r^{n+1}}{(n+1)!}$ where $x \in [x_0 - r, x_0 + r]$ and $|f^{(n)}(x)| \leq M_n$ on $[x_0 - r, x_0 + r]$

Taylor Expansion: Scalar Function of a Single Variable

Examples of Taylor expansion of common elementary functions at 0:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots$$

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots$$

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \ldots$$

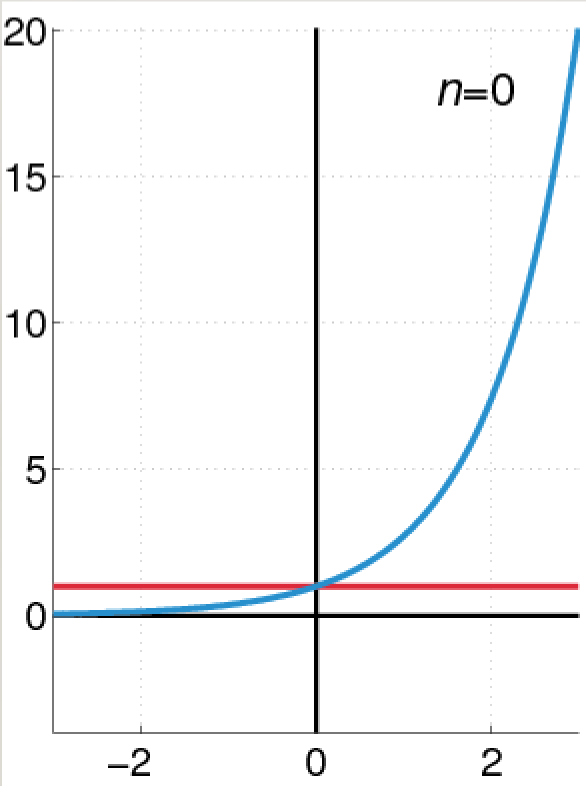
$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + \ldots \quad \text{for} \quad |x| < 1$$

$$\ln(1-x) = -x - \frac{x^2}{2} - \frac{x^3}{3} - \ldots \quad \text{for} \quad |x| < 1$$

Taylor Expansion: Scalar Function of a Single Variable

Examples of Taylor expansion of common elementary functions at 0:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots$$



Taylor Expansion: Scalar Function of Many Variable

Taylor series can be generalized to functions of more than one variable as

$$\sum_{k_1=0}^{\infty}\cdots\sum_{k_n=0}^{\infty}\frac{\partial^{k_1}}{\partial x_1^{k_1}}\cdots\frac{\partial^{k_n}}{\partial x_n^{k_n}}\frac{f(y_1,\ldots,y_n)}{k_1!\cdots k_n!}(x_1-y_1)^{k_1}\cdots(x_n-y_n)^{k_n}$$

Example: second-order Taylor series for a function of two variables:

$$f(x_1,x_2)= f(y_1,y_2)+\frac{\partial f}{\partial x_1}(y_1,y_2)(x_1-y_1)+\frac{\partial f}{\partial x_2}(y_1,y_2)(x_2-y_2)+$$

$$+\frac{1}{2!}\left[\frac{\partial^2 f}{\partial x_1^2}(y_1,y_2)(x_1-y_1)^2+\frac{\partial^2 f}{\partial x_1 \partial x_2}(y_1,y_2)(x_1-y_1)(x_2-y_2)+\frac{\partial^2 f}{\partial x_2^2}(y_1,y_2)(x_2-y_2)^2\right]$$

Vector notation:

$$f(x)= f(y)+\nabla f(y)^T(x-y)+\frac{1}{2!}(x-y)^T\nabla^2 f(y)(x-y)+\ldots$$

Approximation of Derivatives Using Finite Differences (Scalar Function of Single Variable)

Forward difference

$$\Delta_h f(x) = f(x+h) - f(x)$$

Backward difference

$$\nabla_h f(x) = f(x) - f(x-h)$$

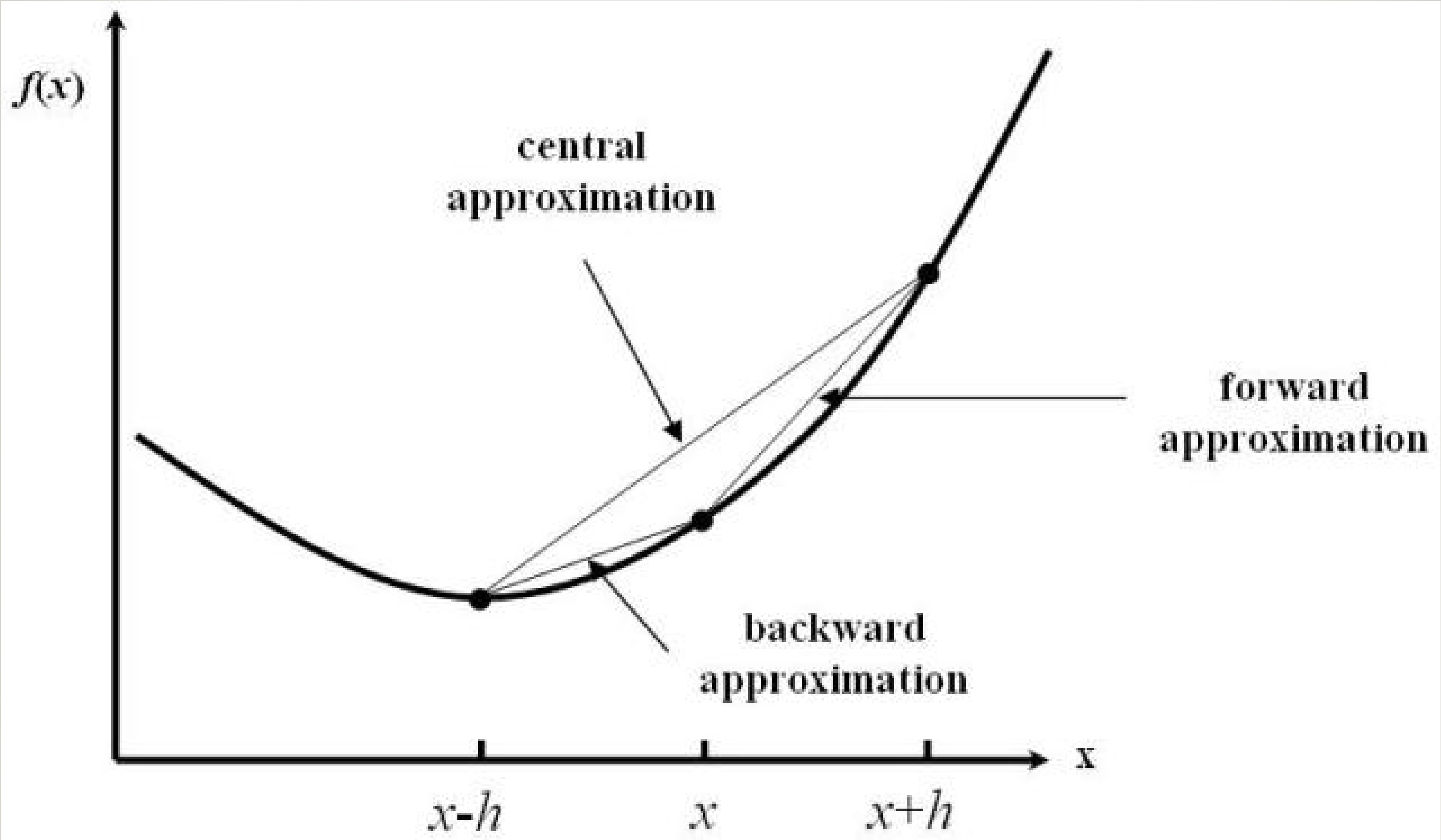
Central difference

$$\delta_h f(x) = f(x+0.5h) - f(x-0.5h)$$

We have

$$f'(x) \approx \frac{\Delta_h f(x)}{h} \approx \frac{\nabla_h f(x)}{h} \approx \frac{\delta_h f(x)}{h}$$

Approximation of Derivatives Using Finite Differences (Scalar Function of Single Variable)



central difference offer better accuracy at the cost of more function evaluation(s)

Approximation of Derivatives Using Finite Differences (Scalar Function of Single Variable)

Error of approximating derivative with finite differences

$$\left| f'(x) - \frac{\Delta_h f(x)}{h} \right| = O(h)$$

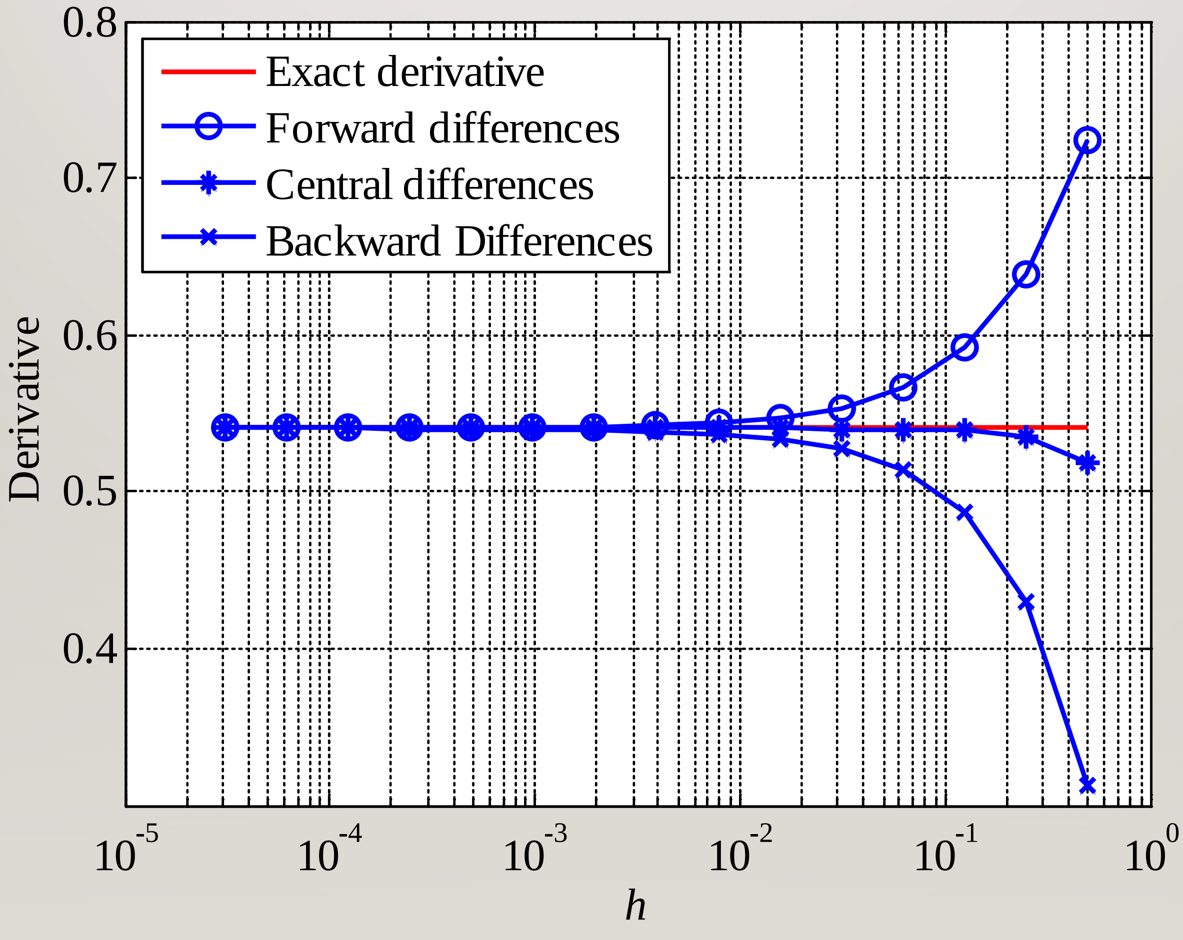
$$\left| f'(x) - \frac{\nabla_h f(x)}{h} \right| = O(h)$$

$$\left| f'(x) - \frac{\delta_h f(x)}{h} \right| = O(h^2)$$

i.e., the error is proportional to h or h^2 in case of central differences (note that central differences are computationally more expensive than forward or backward ones)

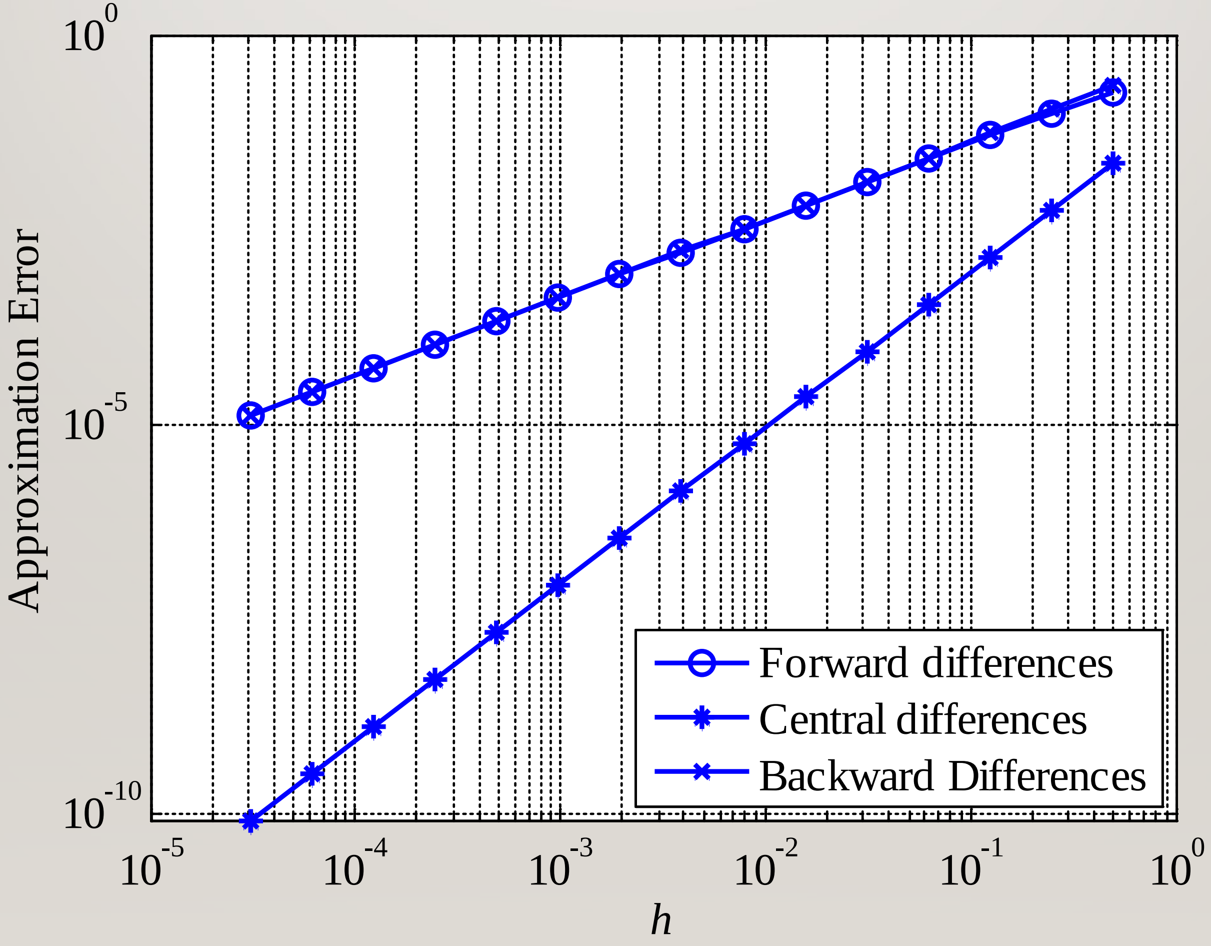
Example: Derivative of $f(x) = \sin(x)$ at x=1 Calculated using Forward, Backward and Central Differences

Estimated derivative vs. finite difference step h; exact value is 0.5403



Example: Derivative of $f(x) = \sin(x)$ at x=1 Calculated using Forward, Backward and Central Differences

Approximation error versus finite difference step *h*



Approximation of Higher-Order Derivatives

Finite differences can be generalized to higher orders

$$\Delta_h^n f(x) = \sum_{i=0}^{n} (-1)^i \binom{n}{i} f(x + (n-i)h)$$

$$\nabla_h^n f(x) = \sum_{i=0}^{n} (-1)^i \binom{n}{i} f(x - ih)$$

$$\delta_h^n f(x) = \sum_{i=0}^{n} (-1)^i \binom{n}{i} f(x + (n/2 - i)h)$$

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!} \quad \text{for } \; 0 \le k \le n,$$

Approximation of Higher-Order Derivatives

Approximation of higher-order derivatives with finite differences:

$$\frac{d^n f}{dx^n}(x) = \frac{\Delta_h^n f(x)}{h^n} + O(h)$$

$$\frac{d^n f}{dx^n}(x) = \frac{\nabla_h^n f(x)}{h^n} + O(h)$$

$$\frac{d^n f}{dx^n} = \frac{\delta_h^n f(x)}{h^n} + O(h^2)$$

Example: approximation of second-order derivative with differences

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

Vector and Matrix Related Matlab Functions:

Matrix multiplication is realized in Matlab using operation *
(use .* in order to perform component-wise multiplication)

Operation $A\backslash b$ is equivalent to $A^{-1}b$

There is a large number of built-in matrix-related commands in Matlab; here is just a few examples of most commonly used ones

```
norm      % calculates norm of a vector/matrix
rank      % calculates rank of a matrix
det       % calculates determinant of a square matrix
inv       % calculates inverse of a square matrix
eig       % calculates eigenvectors and eigenvalues of a matrix
sqrtm     % calculates square root of a matrix
expm      % calculates exponent of a matrix
logm      % calculates logarithm of a matrix
linsolve  % solves system(s) of linear equations
```

Bibliography

Any calculus/linear algebra textbook

Exercise 1: Gram-Schmidt Procedure

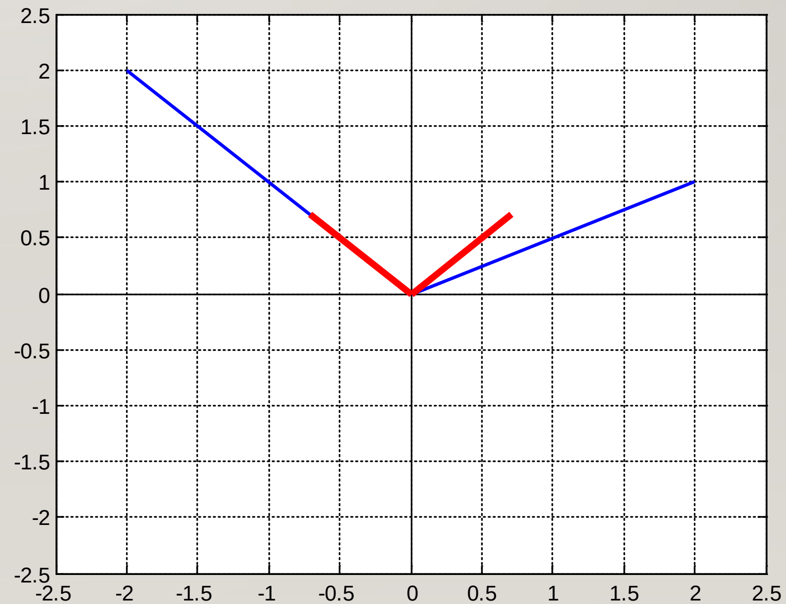
Implement Gram-Schmidt orthonormalization procedure: given linearly independent vectors v_1 to v_k, find orthonormal vectors e_1 to e_k as follows:

$$e_1 = v_1 / \| v_1 \|$$

$$e_j = \frac{t_j}{\| t_j \|} \quad \text{where} \quad t_j = v_j - \sum_{i=1}^{j-1} (v_j^T e_i) e_i, \quad j = 2, \ldots, k$$

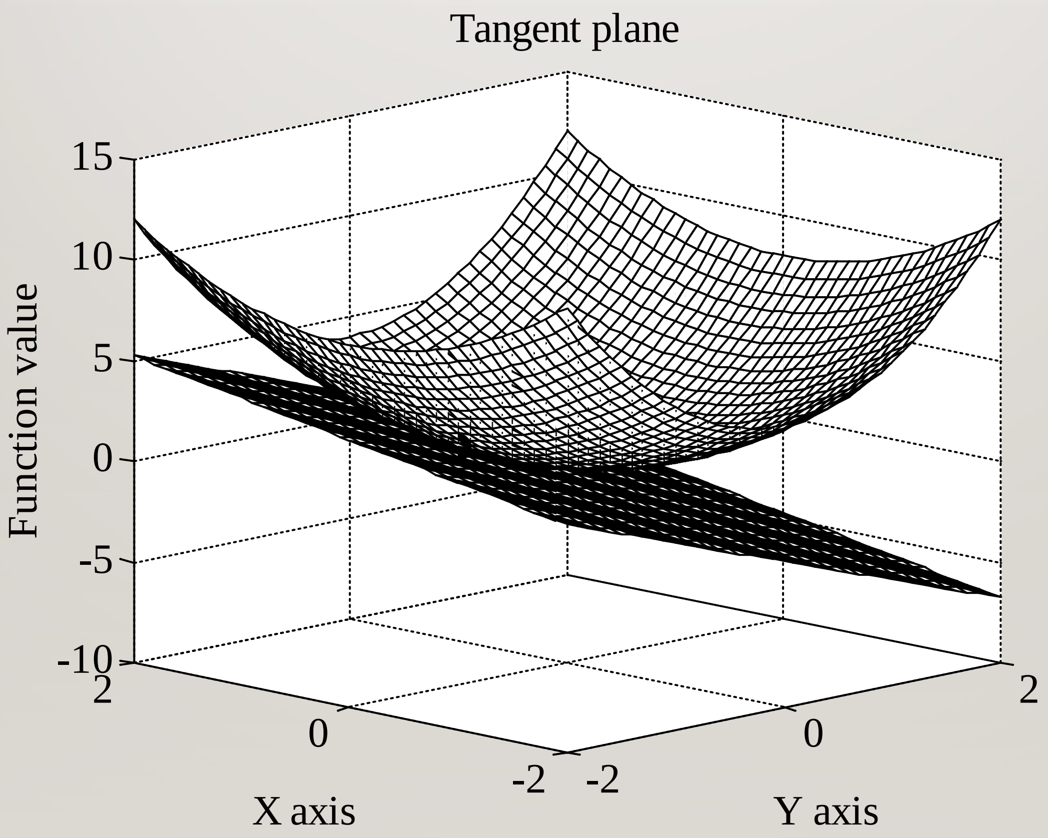
Test the procedure for selected examples. Visualize the procedure for k = 2.

<u>Example</u>: Vectors v_1 = [-2 2]^T and v_2 = [2 1]^T (blue lines). Orthonormal vectors e_1 = [-0.7071 0.7071]^T and e_1 = [0.7071 0.7071]^T (red lines) obtained using Gram-Schmidt procedure:



Exercise 2: Tangent Plane

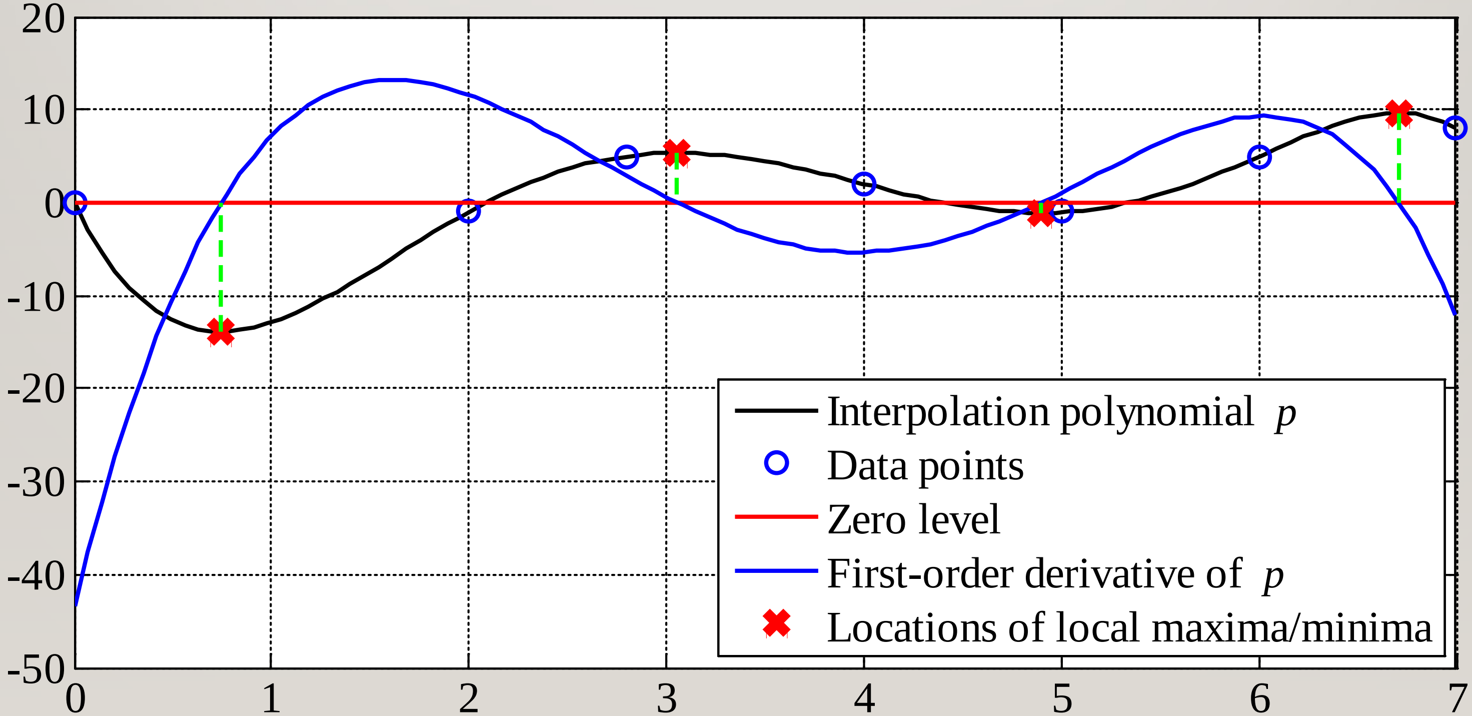
Implement Matlab function that plots a paraboloid $x^2 + y^2$ as well as a tangent plane established at point $[0.5\ -0.5]^T$



Exercise 3: Polynomial Interpolation

Write a Matlab function that finds a polynomial interpolating the data set {(xi,yi)}, i =1,…,n, finds all local minima and maxima of this polynomial and plots both the polynomial, its first-order derivative, input data set as well as locations of the minima/maxima.

Example: Visualization for the data set {(0,0),(2,-1),(2.8,5),(4,2),(5,-1),(6,5),(7,8)}.



Exercise 4: Numerical Methods for ODEs

First-order ordinary differential equation can be written in the form of $y'(t) = f(t,y(t))$ with the initial condition $y(t_0) = y_0$.

Write Matlab functions that solve this equation using Euler's method ($y_{n+1} = y_n + h \cdot f(t_n,y_n)$, where $t_{n+1} = t_n + h$, and h is the method's step) and Adams-Bashforth method ($y_{n+2} = y_{n+1} + 1.5 \cdot h \cdot f(t_{n+1},y_{n+1}) - 0.5 \cdot h \cdot f(t_n,y_n)$, with t_n and h as before).

Test your functions by solving two equations: $y'(t) = -y + 3\cos(3t) \cdot \exp(-t)$, $y(0) = 0$ (exact solution: $y(t) = \sin(3t) \cdot \exp(-t)$), and $y'(t) = y$, $y(0) = 1$ (exact solution $y(t) = \exp(t)$). Plot the numerical and exact solutions for $0 \leq t \leq 5$. Consider $h = 0.2$, 0.1, 0.05, and 0.01.

Example: Solutions to equation $y'(t) = -y + 3\cos(3t) \cdot \exp(-t)$, $y(0) = 0$, for $h = 0.1$: